Idle VMs detection using Condor

Giovanni Franzini

FermiCloud vs. FermiGrid

- FermiCloud → for users that need a computer to work, send email, etc.
- FermiGrid → scientific computing. Resources for compute-intensive jobs.
- An idle VM on FermiCloud occupies resources that can be utilized to run jobs belonging to FermiGrid system.
- A mechanism for idle VMs detection is required.
- Aim of the project:
  - Use Condor tools for detecting idle VMs into FermiCloud.
  - Inform OpenNebula about idle VMs and suspend them.
  - Replace the suspended VMs with FermiGrid worker node to execute jobs in queue.

Condor ClassAd mechanism

- Condor uses ClassAd to represent the characteristics and constraints of machines and jobs into the system.
- ClassAd attributes are keep updated by `condor_startd` daemon (which runs only on worker nodes).
- The `condor_status` command allows users to view ClassAd attributes.
- Users can also define new attributes into Condor's configuration files, combining pre-existent attributes.

How can I use ClassAd?

- Some of the ClassAd attributes are useful in order to detect if a VM is idle:
  - `KeyboardIdle / KeyboardBusy`
  - `CPUIdle / CPUBusy`
  - `ConsoleIdle / ConsoleBusy`
  - `NonCondorLoadAvg`
  - ...
- A new attribute `IsVMIdle` can be defined. This will be used to understand if a VM is idle or not.
- For example it can be defined as follows:

```
IsVMIdle = KeyboardIdle > 24 * $(HOUR) && \
CPUIdle > 12 * $(HOUR)
```

A possible solution

- Once defined `IsVMIdle` into the local Condor configuration file, it is possible to create a simple script to detect if the VM is idle.
- This script follows two steps:
  - Use `condor_status` to read `IsVMIdle` attribute
  - If `IsVMIdle == TRUE`, send to OpenNebula a request of suspension for the VM
- It will run on every VM periodically (every 60 ~ 120 minutes).

A possible solution

- We do not need a Condor Manager for the pool
  - `condor_status` can query directly the startd daemon running on the VM using the following option:

    `-direct <vm_name>`
  - worker nodes can be “isolated”. No Condor network configuration is required.
- All the VMs within FermiCloud can be simple worker nodes.
- They execute only `condor_master` and `condor_startd` daemons and they do not need to communicate to any collector or manager.

The script

```
[…]
vm_name=$(hostname)
idle_vm=$(condor_status -direct $vm_name \
      -const “IS_IDLE == TRUE” \
      -format “%s\n” Machine
      | grep $vm_name)
if [ “$idle_vm” = “$vm_name” ]
then
      <suspend_request_to_OpenNebula>
fi
```

Next steps

- List of all useful ClassAd attributes for idle detection
- Define a Condor policy for identifying an idle VM
- Understand how the script can communicate with OpenNebula
    - REST API?
    - ???